commercially exploited the invention. It is respectfully submitted that this rejection is improper.

According to the CAFC,

> If a use is experimental, even though not secret, public use is negated. If the challenged use or sales activities were associated with primarily experimental procedures conducted in the course of completing the invention, as the particular invention may require, a 35 USC §102(b) bar does not vest. Testing by the public, in public view is not "public use" if the purpose is primarily experimental.

Baker Oil Tools Inc. v. Geo Van Inc., 4 USPQ2d 1210 (CAFC 1987) (emphasis added).

The CAFC also noted that the period of experimental use continues until after the inventor "conducts tests needed to convince himself that the invention is capable of performing its intended purpose in its intended environment. Manville Sales Corp. v. Paramount Systems Inc., 16 USPQ2d 1587 (CAFC 1990, quoting Gould Inc. v. United States, 198 USPQ 156 (Ct. Cl. 1978)) (emphasis added).

Any use of the instant invention prior to May 29, 1992, was primarily experimental. As is well known in the art, the most effective way to test a computer system is to use it in as close to actual conditions as possible, preferably in the use environment itself, as was done in the testing of the present invention. Following the CAFC, since the instant invention required testing in an actual use situation, a 35 USC §102(b) bar cannot vest. Accordingly, it is respectfully submitted that the rejection under 35 USC §102(b) based on prior use and commercial exploitation is

not proper. All use of the invention was in-house and subject to secrecy agreements. The claimed invention has not been commercially exploited.

The Examiner rejects claims 1-19 under 35 USC §102(b) as being clearly anticipated by Perkins et al. This rejection is respectfully traversed.

Perkins describes a way of writing code. The applicant invented a system which automatically writes code for the user. The user does not need to understand how to write code or loops. The system compiles the decisional rules, the system parses the condition; the system provides continuing interactive evaluations; the system resumes processing after the condition is fulfilled.

Perkins describes the opposite: the user writes code which is compiled; the user writes code which parses the condition; the user writes code which provides continuing interactive evaluations; the user writes code which performs the task when the condition is fulfilled; the user writes code which resumes processing after the condition is fulfilled.

Using the technique of Perkins, one cannot enter a command such as "when. . .then" as code into the computer. Instead, the user must write code which accomplishes the function of "when. . .then". To do this, the user must understand computer programming and be able to write loops and nested loops if there are many conditions.

The user must be able to keep track of nested loops,

must exercise configuration control when changes are made, must trouble shoot and debug the programs written -- in short, must do all the things that computer programmers do.

The applicant's invention allows the user to enter a "when. . . then. . ." statement and the claimed system takes care of everything else. In this way, an astronaut on board the space shuttle who has no computer programming experience can enter the command "When T = T_{launch}^{-30}, then ignite rocket engine" and the claimed system takes care of everything else.

The invention has proved, through experimental use, to enable those not skilled in the computer arts to be able to enter commands in the way humans actually think.

Humans do not naturally think in terms of "If. . .then. . ." loops. Four years of computer science education is required before computer programmers can "think" this way.

Instead, humans think in terms of "when it is 12:00, I'll eat lunch". To program this, an "if. . .then. . ." statement must be written and a loop written to keep checking whether or not it is now 12:00.

The applicant's invention eliminates this type of coding by the user. The system writes the necessary code automatically for the user.

Applicant claims a method for automatically evaluating a decisional rule containing a task and a condition which must be fulfilled before the task can be performed and for

automatically performing the task when the condition is fulfilled, comprising entering the decisional rule into computing means; compiling the decisional rule to parse the condition; providing automatic and continuing interactive evaluations of whether the condition is fulfilled until the condition is fulfilled once; automatically performing the task when the condition is fulfilled once; and resuming further processing only after the condition is fulfilled once.

In contrast, Perkins discloses a machine code program which evaluates the occurrence of a condition and performs a task if the condition occurs. The evaluation of the occurrence of the condition is repeated over and over again until the condition occurs, at which time the task is preformed. The loops required to perform the recursive evaluations are part of this programming. Although, in Fig. 7, "WHEN:" and "THEN:" are used to illustrate the processing which occurs, the author's own English translation at the top of the figure clearly shows that the process is an "IF-THEN" process: "If the bearing's temperature exceeds 80° after its thermal model is calculated, [then] a new goal to diagnose the problem is activated."

Furthermore, Perkins does not disclose a system wherein the "WHEN" statement is manually input by the user. Perkins' program is not interactive as all the instructions are in the original program itself. The only input are the specific parameters which the program acts on.

Applicant discloses a WHEN/THEN system which evaluates a decisional rule entered by the user and translates this decisional rule into code in order to allow the user to interface with the system without requiring the user to be able to program the code into the system. See page 6, lines 20-23 and page 7, lines 5-10 of the specification.

Applicant's invention accepts a decisional rule from the user in a "WHEN-THEN" form and then compiles the statement to parse and isolate the condition. See page 8, lines 6-8 and page 11, lines 7-24. The system then performs the necessary operations based on the occurrence of the condition set forth in the decisional rule. This allows those not skilled in the computer science disciplines to enter a command such as "WHEN" instead of "IF" type statements which would require "loops" of programming to enable an executor to recursively evaluate the conditions. See page 17, lines 10-14.

An example of the application of the disclosed system is recited on page 16, line 18 through page 18, line 2.

Applicant claims a system which allows a user to input decisional rules which are then compiled and parsed to determine the condition to be evaluated, as opposed to Perkins, who discloses a pure computer program. The only inputs are the data to be acted upon. If the user wishes to change the condition to be evaluated, he must rewrite the program, including writing loops to change the processing which occurs. Therefore, it is respectfully submitted that

applicants' invention is not anticipated by Perkins.

Since this is a Response to a Final Office Action, no new issues have been raised and it is believed that the application is in better form for appeal.

Each of Examiner's rejections has been addressed or traversed. Accordingly, it is respectfully submitted that the application is in condition for allowance. Early and favorable action is respectfully requested.

If for any reason this RESPONSE is found to be INCOMPLETE, or if at any time it appears that a TELEPHONE CONFERENCE with counsel would help advance prosecution, please telephone the undersigned or his associate, Joseph S. Iandiorio, collect in Waltham, Massachusetts, (617)890-5678.

Respectfully submitted,

Kirk Teska
Reg. No. 36,291